MEYERTONS
HOOD
KIVLIN
KOWERT
& GOETZEL
A PROFESSIONAL CORPORATION
PATENTS, TRADEMARKS, COPYRIGHTS & UNFAIR COMPETITION

700 LAVACA, SUITE 800
AUSTIN, TEXAS 78701
TELEPHONE (512) 853-8800
FACSIMILE (512) 853-8801

FAX

To: Examiner Wu **From: Stephen J. Curran, Patent Agent**

Fax: 571-273-3776 **Pages: 9, including cover page**

Phone: **Date: March 18, 2008**

Re: SN 10/727,477 (5681-72500) **Phone: 512-853-8843**

Pursuant to our conversation, please find attached a Proposed Amendment for the above-identified patent application.

Thank you.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 10/727,477	§	Examiner:	Wu, Qing Yuan
Filed: December 4, 2003	§	Group/Art Unit:	2195
Inventor(s):	§	Atty. Dkt. No:	5681-72500
Welbon et al.	§	Conf. No.	6232
	§		
	§		
	§		
Title: PROCESSOR AND METHOD FOR SUPPORTING COMPILER DIRECTED MULTITHREADING MANAGEMENT	§		
	§		
	§		
	§		
	§		

PROPOSED AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This paper is a proposed amended claim set, in response to a telephone request by the Examiner. This amendment will be entered as an Examiner's amendment.

IN THE CLAIMS:

Please cancel claim 6 without prejudice or disclaimer as to the subject matter contained therein.

Please amend the claims as shown in the following claims listing.

1. (Currently amended) A processor comprising:
an execution unit configured to execute one or more threads; and
a detection unit coupled to the execution unit and configured to detect whether a given thread includes a specific identifier;
wherein said execution unit is further configured to selectively continue execution of said given thread depending upon whether said detection unit detects said specific identifier;
wherein said execution unit is configured to suspend execution of said given thread and to execute a different thread in response to receiving a global execution parameter; [[and]]
wherein in response to said detection unit detecting said specific identifier, said execution unit is configured to override said global execution parameter and to continue execution of said given thread; and
a priority designation unit coupled to said detection unit and configured to assign a priority level to said given thread depending upon an execution environment in response to said detection unit detecting said specific identifier.

2. (Previously presented) The processor of claim 1, wherein in response to said detection unit detecting a different identifier, said execution unit is configured to suspend execution of said given thread and to execute a different thread.

3. (Previously presented) The processor of claim 2, wherein in response to said detection unit detecting that said given thread does not include said different identifier, said executing unit is configured to continue execution of said given thread.

4-6. (Cancelled)

7. (Currently amended) The processor of claim [[6]] 1, wherein in response to said given thread having a priority level lower than said different thread, said execution unit is configured to suspend execution of said given thread and to execute said different thread with a higher priority level

8. (Original) The processor of claim 7, wherein in response to said given thread having a priority level the same as or higher than said different thread, said execution unit is configured to continue execution of said given thread.

9. (Previously presented) The processor of claim 1, wherein said specific identifier is a unique instruction.

10. (Previously presented) The processor of claim 1, wherein said specific identifier is a flag including one or more unused bits of any instruction of said given thread.

11. (Previously presented) The processor of claim 1, wherein said specific identifier is information associated with any instruction of said given thread.

12. (Currently amended) A method comprising:

executing one or more threads;

detecting whether a given thread includes a specific identifier;

selectively continuing execution of said given thread depending upon whether said specific identifier is detected;

suspending execution of said given thread and executing a different thread in response to receiving a global execution parameter; [[and]]

overriding said global execution parameter and continuing execution of said given thread in response to detecting said specific identifier; and

assigning a priority level to said given thread depending upon an execution environment in response to detecting said specific identifier.

13. (Previously presented) The method of claim 12, wherein in response to detecting a different identifier, suspending execution of said given thread and executing a different thread.

14. (Previously presented) The method of claim 13, wherein in response to detecting that said given thread does not include said different identifier, continuing execution of said given thread.

15-16. (Cancelled)

17. (Currently amended) A method of generating low-level instructions executable by a processor, said method comprising:

providing a computer program including high-level programming instructions;

detecting whether an indicator is included within said computer program; and

in response to detecting said indicator, generating a low-level instruction having a specific identifier corresponding to said indicator, wherein said low-level identifier is configured to cause said processor to selectively continue execution of a given thread by:

suspending execution of said given thread and executing a different thread in response to receiving a global execution parameter; [[and]]

overriding said global execution parameter and continuing execution of said given thread in response to detecting said specific identifier; and

assigning a priority level to said given thread depending upon an execution environment in response to detecting said specific identifier.

18. (Original) The method of claim 17, wherein said indicator is a compiler directive.

19. (Original) The method of claim 17, wherein said indicator is an assembly language subroutine.

20. (Original) The method of claim 17, wherein said indicator is a unique high-level instruction.

21. (Currently amended) A machine-readable medium comprising program instructions, wherein said program instructions are executable by a processor to:

execute one or more threads;

detect whether a given thread includes a specific identifier;

selectively continue execution of said given thread depending upon whether said identifier is detected;

suspend execution of said given thread and executing a different thread in response to receiving a global execution parameter; [[and]]

override said global execution parameter and continuing execution of said given thread in response to detecting said specific identifier; and

assign a priority level to said given thread depending upon an execution environment in response to detecting said specific identifier.

22. (Currently amended) A processor comprising:

means for executing one or more threads;

means for detecting whether a given thread includes a specific identifier;

means for selectively continuing execution of said given thread depending upon whether said specific identifier is detected;

means for suspending execution of said given thread and executing a different thread in response to receiving a global execution parameter; [[and]]

means for overriding said global execution parameter and continuing execution of said given thread in response to detecting said specific identifier; and

means for assigning a priority level to said given thread depending upon an execution environment in response to detecting said specific identifier.

REMARKS

Claims 1-22 were pending in the present application. Claim 6 has been cancelled. Claims 1, 12, 17, 21, and 22 have been amended to include the features previously recited in claim 6. Claim 7 was amended to correct claim dependency. Accordingly, claims 1-3, 7-14, and 17-22 are now pending in the application.

Examiner telephone summary

On March 17, the Examiner telephoned Stephen J. Curran, Agent for Applicant (hereinafter "Applicant"). The Examiner indicated claim 6 is allowable, and that if the features of claim 6 were amended into the independent claims, the claims would be allowable. In addition, in response to an objection to claim language in dependent claim 2 not being supported in the specification, Applicant pointed out passages in the specification that support the claim language. Applicant agreed to amend the claims and submit a proposed amendment via facsimile.

Accordingly, to expedite allowance, Applicant has amended the claims as shown in the above claims listing.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-72500/SJC.

Respectfully submitted,

/ Stephen J. Curran /
Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: March 18, 2008